UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,392	02/06/2004	Jong-Ky Lee	P56960	5019

7590 04/01/2008

Robert E. Bushnell
Suite 300
1522 K Street, N.W.
Washington, DC 20005

EXAMINER	
ZHANG, SHIRLEY X	
ART UNIT	PAPER NUMBER
2144	

MAIL DATE	DELIVERY MODE
04/01/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/772,392	LEE, JONG-KY	
	Examiner	Art Unit	
	SHIRLEY X. ZHANG	2144	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *06 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This final office action is prepared in response to the amendments and arguments the applicant filed on February 06, 2008 in reply to the non-final office action mailed on October 12, 2007.

Claims 10-18 have been cancelled;

Claims 1-3 have been amended;

Claims 1-9 are now pending;

Response to Arguments

Applicant's arguments and amendments filed on February 06, 2008 have been carefully considered but are not deemed fully persuasive.

Applicant's arguments are deemed moot in view of the following new grounds of rejection as explained here below, necessitated by Applicant's substantial amendments to the claims which significantly affected the scope thereof, i.e., by amending the independent claim 1, the amendments have changed the scope of claim 1 and its dependent claims, and will require further search and consideration.

Accordingly, THIS ACTION IS MADE FINAL. See MPEP 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

1. It is acknowledged that applicant's formal drawings submitted on February 06, 2004 is accepted by the Examiner.

2. **The 35 U.S.C. 101 rejections of claims 1-3 are withdrawn** after a careful review of the amended claims.

3. **The 35 U.S.C. 103 rejections of claims 1-9 are maintained.** The examiner has carefully considered applicant's amendments and arguments regarding claims 1-3 but respectfully disagrees for the following reasons.

Applicant states that claims 1 and 4 teach a content frame made up of or having dynamic HTML, which limitation is not taught by either Ditmer et al. (US 6,473,407) or Dorland et al. (US 2003/0028577), while the prior art reference Ditmer et al. only discloses Java-based COApp and COApplet.

The examiner respectfully disagrees with the applicant because the Java-based COApplet disclosed by Ditmer et al. extends the Applet class and may be launched by the browser from an HTML <Applet> tag. It is in the knowledge of one of ordinary skill in the art that Java Applet is embedded in a HTML page, delivered in the form of Java byte-code, and run in a Web browser using a Java Virtual Machine (JVM) to generate dynamic contents without a separate loading program.

Accordingly, the examiner considers "an HTML page with Java applets using the <applet> tag" as a "Dynamic HTML" based on its definition provided by the applicant in section 2 "Description of Related Art" of the specification on page 2, paragraph [0005]. Therefore, an HTML page that contains CoApplet with an <Applet> tag as disclosed by Ditmer et al. is equivalent to the dynamic HTML content frame recited in claims 1 and 4.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 39U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

4. **Claims 1, 3 and 10-13** are rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer et al. (U.S. Patent 6,473,407, hereinafter "**Ditmer**"), in view of Dorland et al. (U.S. Application Publication 2003/0028577, hereinafter, "**Dorland**").

As to claim 1, Ditmer teaches a program storage device, readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method of providing a user with alarm information in a data table system, the method comprising:

fixing a title label (Fig. 4 discloses a header frame that displays the title label "networkMCI Interact") of a lightweight alarm manager in a Web browser via a header frame, the alarm manager being applied to a computer connected to a NMS (Network management System) over a network (column 15, lines 50-57 disclose a browser-based

event monitor that is launched from the home page shown in Fig.4, therefore the event monitor is a lightweight alarm manager);

receiving alarm information from the NMS through the network via a data frame (column 16, lines 65-67 and column 17, lines 1-3 disclose that an alarm management object is launched by the GUI client application to handle communications with the multi-tiered server for events or alarms, therefore the alarm management object is equivalent to the data frame recited in the claim. Note that the multi-tiered server includes the Web server/dispatcher/cookie jar server 635, event monitor proxy 640 and the event monitor server 650); and

reading the alarm information being managed in the data frame by a contents frame having dynamic HTML (Hypertext Markup Language) and providing the user with the alarm information via the content frame (column 16, lines 23-55 disclose COApp and COApplet as two different embodiments of the event monitor GUI client application; column 16, lines 35-45 further disclose that a Java-based COAppFrame provides displace space for the event monitor; column 16, lines 51-55 disclose in another embodiment that the Java applet COApplet launched by the browser from an HTML <Applet>tag provides the event monitor with a browser-based display space, therefore both COApp and COApplet are equivalent to the contents frame recited in the claim);

Ditmer does not explicitly teach constructing a data table of the alarm information being managed by the data frame. However, Ditmer teaches in column 5, lines 39-43 that the client software includes additional applications that are directed to front-end services such as the presentation of data in the form of tables and charts.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the contents frame construct a data table of the alarm information being managed by the data frame, if the said alarm information is to be presented in the form of tables, as suggested by Ditmer. Such implementation is in the knowledge generally available to one of ordinary skill in the art at the time the invention was made.

Furthermore, Ditmer does not teach that the alarm manager manages the alarm information in XML format.

However, Dorland teaches managing alarm information in XML format by disclosing in paragraph [0028] that the alarm information requested by the web client is retrieved from the database, converted into XML format, and then passed to the web client via the web server.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dorland's teaching of managing the alarm information in XML format to Ditmer. At the time the invention was made, one would have been motivated to combine Ditmer and Dorland because XML was then already recognized as a general-purpose markup language that facilitated the sharing of structured data across different information systems, particularly over Internet. Such knowledge is also disclosed in paragraph [0007] of Dorland which stated that it would be desirable to provide a common platform to all client applications to receive the events/data in a universal language, i.e. XML.

As to claim 3, the combination of Ditmer and Dorland teaches the method of claim 1.

Ditmer further teaches that the contents frame provides the alarm information that is comprised of {severity} of the alarm, {eventtime} when the alarm is raised, alarm ID{alarm_id}, components of network equipment, {dn}, where the alarm is raised, and contents of the alarm {contents} (column 27, lines 25-34 disclose that the attributes of alarms recorded are severityLevel_, timeStamp_, alarmID_, interfaceID_, and alarmText_, which correspond to the {severity}, {eventtime}, {alarm_id}, {dn} and {contents} recited in the claim, respectively).

5. **Claim 2 is** rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer and Dorland, as applied to claim 1 above, further in view of Flanagan (Chapter 17 of the book "JavaScript: The definitive Guide, 4th Edition", hereinafter "**Flanagan**")

As to claim 2, the combination of Ditmer and Dorland teaches the method of claim 1.

Ditmer further teaches a contents frame that provides a GUI (Graphical User Interface) (column 5, lines 53-62 disclose that the client software provides a reusable and common GUI abstraction).

Furthermore, Ditmer discloses that the client software includes additional applications that are directed to front-end services such as the presentation of data in the form of tables and charts (column 5, lines 39-43).

Neither Ditmer nor the combination of Ditmer and Dorland explicitly teaches a table object of a HTML provided by the Web browser.

However, Flanagan teaches that an HTML table objects is provided by a Web browser to display information (pages 27-29 of chapter 17 disclose the method of

displaying information on a Web browser by loading an XML document, extracting information from it, and dynamically creating an HTML version of that information for display in a HTML table object provided by the browser. See page 28, example 17-9 for more details).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the tables recited by Ditmer as an HTML table object, as taught by Flanagan. One would have been motivated to combine as such for the reason that at the time the invention was made, HTML table object was already a well-defined element for supporting structured display of data such as alarm information on a Web browser, and software tools such as JavaScript were readily available to dynamically edit the table contents.

6. **Claim 4** is rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer et al. (U.S. Patent 6,473,407, hereinafter "**Ditmer**").

As to claim 4, Ditmer teaches a service method of a lightweight alarm manager running in a Web browser (column 15, lines 50-57 disclose a browser-based event monitor that is launched from the home page shown in Fig.4, therefore the event monitor is a lightweight alarm manager), to be applied to a computer connected NMS (Network Management System) through network, the service method comprising the steps of:

receiving a request from a user to use the alarm manager (column 15, lines 55-57 disclose that the user can request to launch the event monitor by clicking the event monitor icon on the home page, as shown in Fig.4);

creating a header frame, a contents frame, and a data frame on the Web browser in the alarm manager, in response to an alarm manager service request from a user (Fig. 4 discloses a **header frame** that displays the title label "networkMCI Interact"; column 16, lines 65-67 and column 17, lines 1-3 disclose that an alarm management object is launched by the GUI client application to handle communications with the multi-tiered server for events or alarms, therefore the alarm management object is equivalent to the **data frame** in the instant application; column 16, lines 23-55 disclose COApp and COApplet as two different embodiments of the event monitor GUI client application; column 16, lines 35-45 further disclose that a Java-based COAppFrame provides displace space for the event monitor; column 16 lines 51-55 disclose in another embodiment that a Java applet COApplet launched by the browser from an HTML <Applet> tag can provide the event monitor with a browser-based display space; therefore COApp and COApplet are the equivalents of the **contents frame;** Furthermore, all the above applications were launched by clicking the event monitor icon on the home page. Note that the multi-tiered server includes the Web server/dispatcher/cookie jar server 635, event monitor proxy 640 and the event monitor server 650);

requesting, at the data frame, that the NMS provides alarm information periodically to the data frame (column 21, lines 52-58 disclose that reports of events may be periodically forwarded based on a customer configurable interval to the client browser application, i.e. the data frame);

managing the alarm information in the alarm manager when the alarm information is received by the data frame (column 13, lines 28-39 disclose a list of alarm management activities carried out by the event monitor);

periodically checking, by the contents frame made up of dynamic HTML, whether the alarm information in the data frame is being properly managed (column 16 lines 51-55 disclose that a Java applet COApplet launched by the browser from an HTML <Applet> tag can provide the event monitor with a browser-based display space, i.e., the contents frame is made up of dynamic HTML; column 17, lines 3-6 disclose that an alarm thread is created to run periodically to poll for current event monitor alarms);

accessing and obtaining, by the contents frame, the alarm information being managed by the data frame (column 17, lines 7-9 disclose that the said alarm thread creates a command to update the display when it receives data back from the web server, i.e. the contents frame is updated with information managed by the data frame);

displaying the alarm information to the user (column 16, lines 39-44 and column 16, lines 52-57 disclose the Java classes COAppFrame and COApplet that provides display space for the alarm information).

Ditmer does not explicitly teach constructing a data table of the alarm information being managed by the data frame. However, Ditmer teaches in column 5, lines 39-43 that the client software includes additional applications that are directed to front-end services such as the presentation of data in the form of tables and charts.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to have the contents frame construct a data table of the alarm information being managed by the data frame, if the said alarm information is to be presented in the form of tables, as suggested by Ditmer. Such implementation is in the knowledge generally available to one of ordinary skill in the art at the time the invention was made.

7. **Claim 5** is rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer as applied to claim 4 above, further in view of Dorland et al. (U.S. Application Publication 2003/0028577, hereinafter, "**Dorland**").

As to claim 5, Ditmer teaches the method of claim 4, wherein the requesting step is comprised of the sub-steps of

requesting the NMS connected to the data frame through the network to provide alarm information periodically (column 17, lines 1-6 disclose that the alarm management object periodically creates an alarm thread to poll for current event monitor alarms);

receiving alarm information from the NMS (column 17, line 7 discloses that the alarm thread receives the data back from the web server); and

managing the received alarm information (column 18, lines 41-45 disclose that the event monitor manages the received alarm information by launching a pre-defined trouble shooting procedure);

Ditmer does not teach that the alarm information is received and managed in XML format.

However, Dorland teaches managing alarm information in XML format by disclosing in paragraph [0028] that the alarm information requested by the web client is retrieved from the database, converted into XML format, and then passed to the web client via the web server.

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Dorland's teaching of managing the alarm information in XML format

to Ditmer. At the time the invention was made, one would have been motivated to combine Ditmer and Dorland because XML was then already recognized as a general-purpose markup language that facilitated the sharing of structured data across different information systems, particularly via the Internet. Such knowledge is also disclosed in paragraph [0007] of Dorland which stated that it would be desirable to provide a common platform to all client applications to receive the events/data in a universal language, i.e. XML.

8. **Claims 6 and 8** are rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer as applied to claim 4 above, in view of Flanagan ("JavaScript: The definitive Guide, 4th Edition, hereinafter "**Flanagan**").

As to claim 6, Ditmer teaches the method of claim 4, wherein the accessing and obtaining, constructing and displaying steps comprise:

obtaining, by the contents frame, the alarm information in XML format from the NMS (column 17, line 7 discloses that the alarm thread receives the data back from the web server);

managing, by the data frame, the received alarm information in the data frame (column 18, lines 41-45 disclose that the event monitor manages the received alarm information by launching a pre-defined trouble shooting procedure);

Ditmer further discloses in column 5, lines 39-43 that the client software includes additional applications that are directed to front-end services such as the presentation of data in the form of tables and charts.

Ditmer does not teach but Flanagan teaches adding a row to a table object, by the contents frame, using attributes of the table object of a HTML provided by the Web browser; and displaying the alarm information being obtained using the table object (Flanagan, chapter 17, pages 27-29, "The Document Object Model" disclose that an HTML table object can be used to display information extracted from an XML document; Flanagan, pages 28, example 17-9 further discloses that JavaScript supports the DOM object HTMLTableElement, which provides the method HTMLTableElement.insertRow() for adding a row to an HTML table object. The HTML page is automatically reloaded after the HTML table object is populated, causing the new alarm information to be displayed).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Flanagan with Ditmer such that the contents frame would use attributes of an HTML table object provided by the Web browser to add a row to the table and display the alarm information. One would have been motivated to combine as such for the reason that HTML table object was already a well-defined element for supporting structured display of data such as alarm information on a Web browser, and software tools such as JavaScript were readily available to dynamically edit the table contents, as taught by Flanagan.

As to claim 8, the combination of Ditmer and Flanagan teaches the method of claim 6.

Neither Ditmer nor the combination of Ditmer and Flanagan specifically teaches checking whether a number of current rows in the table object provided by the Web browser is greater than a predetermined number of rows;

However, it would have been obvious to one of ordinary skill in the art at the time of the invention to check whether the number of current rows in the table object was greater than a predetermined number of rows because it was as simple as comparing two integers.

Furthermore, Ditmer does not teach but Flanagan teaches the step for adding a row by the contents frame comprises sub-steps of:

deleting the oldest record when the number of current rows in the table object provided by the Web browser is greater than the predetermined number; creating a new row in the table object comprising the alarm information read by the contents frame; and displaying the alarm information of the table object when the number of current rows in the table object provided by the Web browser is not greater than the predetermined number of rows to be maintained (Flanagan, book chapter 17, "The Document Object Model" disclose that JavaScript supports the DOM object HTMLTableElement that supports the methods HTMLTableElement.deleteRow() and HTMLTableElement.insertRow() for deleting a row from or adding a row to an HTML table object. The HTML page is reloaded after the HTML table object is populated, causing the new alarm information to be displayed).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Flanagan with Ditmer such that the contents frame would use attributes of an HTML table object provided by the Web browser to delete the oldest record when the number of current rows in the table object provided by the Web browser is greater than the predetermined number, create a new row in the table object comprising the alarm information read by the contents frame, and display the alarm

information of the table object when the number of current rows in the table object provided by the Web browser is not greater than the predetermined number of rows to be maintained.

One would have been motivated to combine as such for the reason that HTML table object was already a well-defined element in HTML for supporting structured display of data such as alarm information on a Web browser, and software tools such as JavaScript were readily available to dynamically edit the table contents, as taught by Flanagan.

9. **Claims 7 and 9** are rejected under 35 U.S.C. 103(a) as being unpatentable over Ditmer and Dorland as applied to claim 5 above, further in view of Flanagan ("JavaScript: The definitive Guide, 4th Edition, hereinafter "**Flanagan**").

As to claim 7, the combination of Ditmer and Dorland teaches the method of claim 5.

Ditmer further teaches that the accessing and obtaining, constructing and displaying steps comprise:

obtaining, by the contents frame, the alarm information in XML format from the NMS;

managing, by the data frame, the received alarm information in the data frame (column 17, line 7 discloses that the alarm thread receives the data back from the web server);

Neither Ditmer nor the combination of Ditmer and Dorland teaches adding a row to a table object, by the contents frame, using attributes of the table object of a HTML

provided by the Web browser; and displaying the alarm information being obtained using the table object.

However, Flanagan teaches in chapter 17 "The Document Object Model" that JavaScript supports the DOM object HTMLTableElement, a table object of a HTML page provided by the Web browser to display information in a structured manner. The HTMLTableElement supports the method HTMLTableElement.insertRow() for adding a row to an HTML table object, and the HTML page is reloaded after the HTML table object is populated, causing the new alarm information to be displayed.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Flanagan with Ditmer and Dorland such that the contents frame would use attributes of an HTML table object provided by the Web browser to add a row to the table and display the alarm information. One would have been motivated to combine as such for the reason that HTML table object was already a well-defined element for supporting structured display of data such as alarm information on a Web browser, and software tools such as JavaScript were readily available to dynamically edit the table contents, as taught by Flanagan.

As to claim 9, the combination of Ditmer, Dorland, and Flanagan teaches the method of claim 7.

Neither Ditmer nor the combination of Ditmer, Dorland, and Flanagan specifically teaches checking whether a number of current rows in the table object provided by the Web browser is greater than a predetermined number of rows;

However, it would have been obvious to one of ordinary skill in the art at the time of the invention to check whether the number of current rows in the table object is greater

than a predetermined number of rows because this is as simple as comparing two integers.

Flanagan further teaches the step for adding a row by the contents frame comprises sub-steps of:

deleting the oldest record when the number of current rows in the table object provided by the Web browser is greater than the predetermined number; creating a new row in the table object comprising the alarm information read by the contents frame; and displaying the alarm information of the table object when the number of current rows in the table object provided by the Web browser is not greater than the predetermined number of rows to be maintained (Flanagan, book chapter 17, "The Document Object Model" disclose that JavaScript supports the DOM object HTMLTableElement that supports the methods HTMLTableElement.deleteRow() and HTMLTableElement.insertRow() for deleting a row from or adding a row to an HTML table object. The HTML page is reloaded after the HTML table object is populated, causing the new alarm information to be displayed).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine Flanagan with Ditmer such that the contents frame would use attributes of an HTML table object provided by the Web browser to delete the oldest record when the number of current rows in the table object provided by the Web browser is greater than the predetermined number, create a new row in the table object comprising the alarm information read by the contents frame, and display the alarm information of the table object when the number of current rows in the table object

provided by the Web browser is not greater than the predetermined number of rows to be maintained.

One would have been motivated to combine as such for the reason that HTML table object was already a well-defined element in HTML for supporting structured display of data such as alarm information on a Web browser, and software tools such as JavaScript were readily available to dynamically edit the table contents, as taught by Flanagan.

Conclusion

THIS ACTION IS FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SHIRLEY X. ZHANG whose telephone number is

(571)270-5012. The examiner can normally be reached on Monday through Friday 7:30am - 5:00pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/S. X. Z./
Examiner, Art Unit 2144
03/14/2008

/William C. Vaughn, Jr./
Supervisory Patent Examiner, Art Unit 2144